

The Game Room: “Draw It or Lose It”

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc115077317)

[**Table of Contents 2**](#_Toc115077318)

[**Document Revision History 2**](#_Toc115077319)

[**Executive Summary 3**](#_Toc115077320)

[**Requirements 3**](#_Toc115077321)

[**Design Constraints 3**](#_Toc115077322)

[**System Architecture View 3**](#_Toc115077323)

**Domain Model 4**

**Evaluation 5**

**Recommendations 9**

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 04/21/2024 | Amos Roland | Integrating additional information to enrich the software architecture of the "Draw It or Lose It" game. |

**Instructions**

Fill in all bracketed information on page one (the cover page), in the Document Revision History table, and below each header. Under each header, remove the bracketed prompt and write your own paragraph response covering the indicated information.

## [Executive Summary](#_sbfa50wo7nsh)

The project aimed to create a web-based version of Gaming Room's well-known Android game, "Draw It or Lose It," that could be played on different platforms. This would help Gaming Rooms to distribute their gaming applications across various operating systems. The Client wanted a gaming experience similar to the "Win, Lose, or Draw" game show from the 80s.

Creative Technology Solutions, acting as technical consultants, spearheaded the development of "Draw It or Lose It," a web-based game for customers (Gaming Room). The game was designed to cater to numerous teams, with several players in each, and was intended to expand beyond its current Android platform to reach a wider audience. The game was to have a unique identity for both the game itself and each team.

To meet the Client's requirements, Creative Technology Solutions is committed to providing the necessary technological resources, work environment, and project parameters. This would ensure the development of a web-based version aligned with five essential software specifications.

## Requirements

*The requirements for "Draw It or Lose It" include supporting multiple teams with multiple players, ensuring distinct team and game names, maintaining a single game version at any given time, enabling cross-platform compatibility beyond Android, and providing sufficient server-side storage for the client's photo library.*

## [Design Constraints](#_2et92p0)

1. The game will feature multiple teams, each comprising several players.

2. Stringent checks will be implemented to ensure unique team and game names.

3. Only one version of the game will be available at any given time.

4. The game will be compatible with various platforms beyond Android.

5. Sufficient storage space will be allocated, particularly on the server side, to accommodate the client's extensive stock photo library.

6. The game will be developed using the appropriate programming language, with Java being the preferred choice to align with the client's existing Android app codebase.

7. Security measures will be implemented to safeguard login credentials, with a focus on utilizing two-factor or multi-factor authentication for enhanced security.

8. The server infrastructure will be designed to handle concurrent requests efficiently, ensuring seamless gameplay experiences for the expanding, multi-platform player base.

9. A comprehensive player management system will be developed to oversee player, game, and team management aspects of the game.

## [System Architecture View](#_ilbxbyevv6b6)

Please note: There is nothing required here for these projects, but this section serves as a reminder that describing the system and subsystem architecture present in the application, including physical components or tiers, may be required for other projects. A logical topology of the communication and storage aspects is also necessary to understand the overall architecture and should be provided.

## [Domain Model](#_8h2ehzxfam4o)

The technological implementation of the "Draw It or Lose It" game is illustrated in the diagram below. The diagram depicts seven game classes that collectively create the desired environment outlined by The Gaming Room. The Game, Team, and Player classes are directly derived from the Entity class, which serves as the superclass.

The Game\_service accesses the Player class along with the Game and Team classes. These subclasses share common properties, such as name, ID, actions, getters, and toString methods inherited from the Entity base class. Players can interact within the game to store and display data about one another. Teams, as per the client's requirement, can have multiple players assigned to them, establishing a relationship between Player and Team classes. Each team maintains a private list of players and offers a public method to add new participants.

In the "Draw It or Lose It" game, the Game class maintains a private list of teams, establishing a connection between Game and Team classes. The "add\_team" method ensures each team name is unique, while a modified toString method notifies users if their preferred identity is already in use.

Considering the singleton nature of the Game\_service class, only one instance of the game can be active in memory at any given time. A Singleton tester class with the test\_singleton operation is utilized in the Program Driver class to verify this functionality. With successful tests, all user software specifications for the game-based app are met effectively, utilizing common design patterns like Singleton and Iteration, along with standard object-oriented concepts such as inheritance and overriding.

Additionally, the Program\_driver class and Singleton\_tester are incorporated into the software design. Whenever the program is executed, it employs the Singleton\_tester class within its Program\_driver

method. Both classes share and utilize each other's functionalities seamlessly.

**"The Gaming Room UML diagram. The top of the diagram is labeled as com dot gamingroom. Test boxes are placed in two layers. The first layer has three text boxes and the second layer has four of them. In the first layer, the 'ProgramDriver' textbox points to 'SingletonTester' textbox. The 'ProgramDriver' textbox contains the text 'asterisk main round brackets.' The 'SingletonTester' textbox contains the text 'asterisk testSingleton round brackets.' The arrow between these two text boxes are labeled 'open two angle brackets uses close two angle brackets'. In the second layer, there are 'GameService', 'Game', 'Team', and 'Player' text boxes. The 'GameService' textbox has texts arranged in two layers. The first layer contains games colon List open angle bracket Game close angle bracket, nextGamesId colon long, nextPlayer Id colon long, nextTeamId colon long, and service colon GameService. The second layer contains GameService round brackets, getinstance round brackets colon GameService, addGame open parenthesis name colon String close parenthesis colon Game, getGame open parenthesis id colon long close open parenthesis colon Game, getGame open open parenthesis name colon String close open parenthesis colon Game, getGameCount round brackets colon int, getNextPlayerID round brackets colon long, and getNextTeamId round brackets colon long. The 'GameService' box is connected with the 'Game' textbox with a line labeled 'zero dot dt dot asterisk'.  The 'Game' textbox also contains text in two layers. The first layers contains the text teams colon List open angle bracket Team close angle bracket. The second layer has Game open round bracket id colon long comma name colon String close parenthesis, addTeam open parenthesis name colon String close parenthesis Team, toString round brackets colon String. The 'Game' textbox is connected with the 'Team' textbox with a line labeled 'zero dot dt dot asterisk'. The 'Team' textbox also contains text in two layers. The first layers contains the text players colon List open angle bracket Player close angle bracket. The second layer has Team open parenthesis id colon long comma name colon String close parenthesis, addPlayer open parenthesis name colon String close parenthesis colon Player, and toString round brackets colon String. The 'Team' textbox is connected with the 'Player' textbox with a line labeled 'zero dot dt dot asterisk'. It contains the text Player open parenthesis id colon long comma name colon String close parenthesis and toString round brackets colon String. The 'Game', the 'Team, and the 'Player' boxes point to the 'Entity' textbox in first layer. The 'Entity' textbox contains text in two layers. The first layer has the text id colon long and name colon String. The second layer has Entity round brackets, Entity open parenthesis id colon long comma name colon String close parenthesis, getId round brackets colon long, getName round brackets colon String, toString round brackets colon String.**

## [Evaluation](#_2o15spng8stw)

Using your experience to evaluate the characteristics, advantages, and weaknesses of each operating platform (Linux, Mac, and Windows) as well as mobile devices, consider the requirements outlined below and articulate your findings for each. As you complete the table, keep in mind your client’s requirements and look at the situation holistically, as it all has to work together.

In each cell, remove the bracketed prompt and write your own paragraph response covering the indicated information.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | **Characteristics:**  - Design and server adjustments are facilitated through flexible terminal commands on Mac.  - Mac boasts exceptional technical support options and product offerings.  - The graphical user interface (GUI) on Mac is notably intuitive.  - Mac enables centralized management of desktop and mobile devices, along with various capabilities such as file sharing, backup, calendars, email, blogs, and OSx delivery.  **Advantages:**  - Mac users can easily create features similar to Windows group policies.  - Mac's security features are robust and reliable.  **Disadvantages:**  - Macs are less commonly used for certain web hosting services.  - Many game developers prioritize creating games for Windows OS due to its larger user base, resulting in fewer game options for Mac users. | **Characteristics:**  - Linux offers several versions tailored for server functionality.  - It is an open-source operating system, freely available for modification under the General Public License (GPL).  **Advantage:**  - Linux requires minimal hardware resources and enhances system security, providing robust protection against viruses and online threats.  **Disadvantages:**  - Setting up Linux servers can be more complex and demanding compared to Windows, often requiring manual configuration file editing and command line usage.  - Linux may have a more limited software availability compared to Windows operating systems. | **Characteristics:**  - Windows Server offers features such as virtual memory management, a full desktop graphical user interface, multitasking, and support for various peripheral devices. While it may have a higher implementation cost, Microsoft ensures its functionality on their website.  **Advantages:**  - Efficient server-based delivery of the Windows OS is a standout feature.  - A vast selection of software supports diverse hardware options, and a large user base ensures prompt updates.  - Enhanced internal and external communication capabilities.  **Disadvantages:**  - Windows Server is considered less secure compared to Linux and Mac.  - Closed-source nature requires payment for usage, with updates exclusively from Microsoft.  - Limited flexibility and reliance on Microsoft updates.  - High cost of Windows servers due to operating system licensing fees.  - Greater susceptibility to online threats, viruses, and security vulnerabilities. | **Characteristics:**  - Mobile smartphones have the capability to wirelessly transmit and receive data via Bluetooth, Wi-Fi, or cellular connections, and typically feature user interfaces like touchscreens or keyboards. They also come equipped with built-in data storage and operate using battery power without a direct electrical connection.  **Advantages:**  - Smartphones enable streamlined communication with servers for database queries and data retention.  - They support wireless payment methods.  - Enhance both internal and external communication capabilities.  - Improve network connectivity and offer better service flexibility and efficiency to customers.  **Disadvantages:**  - Integration with server-side systems may require the use of cloud services or physical servers.  - Portable devices, including smartphones, are vulnerable to security attacks, particularly if they contain sensitive corporate data.  - The risk of physical damage or loss due to dropping or misplacement is significant. |
| **Client Side** | The cost of acquiring a Mac is comparable to configuring a Windows system, given that both operating systems are proprietary. Additionally, the time required to become proficient in using a Mac varies depending on the individual's skill level; experienced Mac users may require less time, whereas novices may need more time to familiarize themselves with the platform. Windows and Android OS are popular alternatives for many users, and transitioning to a Mac can be challenging for those who have never used an Apple product before. Proficiency in operating a Mac is essential before achieving optimal performance.  Developing software for Mac customers necessitates developers with significant experience in Mac programming. Some users who are new to Apple products or the Mac OS may find the experience enjoyable and become enamored with the platform, while others may find the costs and inconveniences associated with Mac usage off-putting. Overall, compared to Windows or Linux, using a Mac tends to be more expensive. | Utilizing Linux effectively requires proficiency with the operating system, so it's essential to have someone on the team who is skilled in Linux administration. Even for experienced users, Linux can pose challenges that require time to address. Linux is known for being a cost-effective open-source operating system, but its effectiveness hinges on the user's knowledge and expertise. Due to the relatively limited availability of applications compared to other platforms, Linux users typically require a high level of technical knowledge to navigate and utilize its functionalities.  One of Linux's notable advantages is its faster boot times and efficient time management capabilities. Despite being less popular than Windows and Mac, Linux can come with a high price in terms of the expertise required to harness its full potential.. | Windows users typically adapt to the operating system quickly due to its widespread use and familiarity among many people. The learning curve for Windows is generally shorter, resulting in quicker comprehension and reduced time investment. Similar to Mac, Windows is a closed-source operating system, so the cost of acquisition is comparable to configuring a Windows system. The time required to become proficient in using Windows varies depending on the individual's level of expertise, with experienced users needing less time compared to novices.  Furthermore, because Windows is not open source like Linux, it incurs higher license costs similar to Mac. | Mobile devices are known for their affordability and fast page loading speeds, outperforming other operating systems in terms of speed. The popularity of mobile devices ensures that consumers have access to high-quality technical support when needed, mitigating potential issues. While mobile smartphones may not pose significant cost concerns, managing multiple operating systems and devices may require additional time and resources. However, the user-friendly nature of mobile devices means that proficiency may not be a major obstacle. Overall, while working with various operating systems and mobile devices may require more time, the ease of use and accessibility of mobile devices offset potential challenges. |
| **Development Tools** | Users have the option to utilize JavaScript and the Hypertext Preprocessor (PHP) programming language for various development tasks. Swift, a programming language specifically designed for macOS, iOS, watchOS, and tvOS development, is also widely used.  For Mac OS development, developers can leverage native Integrated Development Environments (IDEs) like Xcode, which offers comprehensive tools and resources for macOS and iOS development. Alternatively, open-source IDEs such as Eclipse and NetBeans can also be utilized for Mac OS development.  Mac OS itself was developed using multiple programming languages, with prominent involvement from C and Objective-C. These languages have played crucial roles in shaping the foundation and functionality of the Mac OS ecosystem. | Linux operating systems are typically developed using a combination of Python, Objective-C, and C programming languages. While Eclipse is commonly associated with Java development, it can also support other languages such as C++. Additionally, various Integrated Development Environments (IDEs) are utilized for Linux development, including Visual Studio, Sublime Text, Atom, and Brackets.  Moreover, developers can also employ the Hypertext Preprocessor (PHP) programming language for certain tasks related to Linux development. PHP is commonly used for web development but can also be utilized in other contexts, including Linux system programming and scripting. | The Windows operating system was developed using a diverse set of programming languages, including C, C++, C#, Java, and HTML/CSS. These languages were instrumental in building various components and functionalities of the Windows platform.  For Windows development, a range of Integrated Development Environments (IDEs) are available, each offering unique features and capabilities. Popular IDEs used for building Windows applications include Visual Studio, Eclipse, NetBeans, JetBrains Rider, and Android Studio. These IDEs provide developers with tools for coding, debugging, and testing Windows software across different programming languages and frameworks. | Smartphone servers are constructed using a combination of programming languages such as C++, Java, and Kotlin. These languages provide the foundation for developing various server-side functionalities and services for smartphones.  For smartphone server development, developers commonly utilize Integrated Development Environments (IDEs) like Android Studio, Eclipse, Visual Studio, and IntelliJ IDEA. These IDEs offer robust tools and features tailored to mobile application development, allowing developers to efficiently build, test, and deploy server-side components for smartphones across different platforms and operating systems. |

## Recommendations

Analyze the characteristics of and techniques specific to various systems architectures and make a recommendation to The Gaming Room. Specifically, address the following:

1. **Operating Platform**: I highly recommend utilizing Windows as the primary operating system for expanding "Draw It or Lose It" to various computing environments. Windows offers seamless integration with the current Android release, simplifying the development process. With tools like Android Studio or cross-platform solutions such as Xamarin or Cordova, setting up your development environment on Windows is straightforward.

Windows Operating System provides an ideal environment for creating games like "Draw It or Lose It" across multiple platforms, as it is widely used in web-based software development. It offers simplicity, security, and affordability, making it an excellent choice for game development. Additionally, Windows' compatibility and portability ensure that the "Draw It or Lose It" game can be efficiently designed and deployed across different devices and platforms.

1. **Operating Systems Architectures**: Windows adopts a two-tiered architecture, consisting of user mode and kernel mode. User mode processes primarily cater to user interaction, influencing various aspects of user experience. On the other hand, kernel mode manages critical system functions such as input/output operations, memory management, networking, hardware management, and low-level routines.

The Windows operating platform comprises several key components, including Configuration and maintenance, Windows Server, User interface, Applications and utilities, File systems, and Core features. User and kernel modes play pivotal roles in managing system resources and interactions.

Kernel mode programs enjoy direct and unrestricted access to system resources, while application programs operate within the confines of user mode. Windows supports both homogeneous multi-processor and single-processor systems, providing a reactive and re-entrant environment for efficient operation.

3. **Storage Management**: Efficiently managing storage capacity is crucial for the "Draw It or Lose It" game, which will incorporate high-definition videos and images. Windows operating system offers built-in tools such as Disk Management and Storage Sense to assist in storage management. Disk Management is primarily used for advanced storage tasks, providing users with sophisticated storage management capabilities.

Additionally, Windows includes utilities like Disk Cleanup and Storage Sense, which help in managing system storage by removing unnecessary files and optimizing storage usage.

For the "Draw It or Lose It" game, implementing a database management system (DBMS) as a storage solution is highly recommended. A DBMS provides a structured and efficient way to organize and store game data, ensuring proper performance and scalability. Windows' high compatibility with various DBMS makes it an ideal platform for implementing such a storage system, ensuring seamless integration and optimal performance.

4. **Memory Management**: Memory management involves the efficient organization and allocation of primary memory resources within a computer system. It ensures that memory space is effectively utilized and allocated to the operating system, programs, and running processes to facilitate their tasks. Windows 11, the latest version of the Windows operating system, features enhanced memory management capabilities aimed at improving memory loading speed and efficiency.

Given that the "Draw It or Lose It" game utilizes high-quality graphics, it requires substantial memory resources. Windows 64-bit architecture is well-suited for such demanding applications, providing ample memory space for efficient operation.

To address the high memory utilization of the "Draw It or Lose It" software, the Windows system employs memory compression techniques to optimize memory usage and enhance system responsiveness. Additionally, creating a database to store all game image data enables quick and efficient access to resources.

When the memory required by the "Draw It or Lose It" software exceeds the available RAM, Windows employs memory paging mechanisms to temporarily store excess data on the hard drive. Users can choose between physical and virtual memory options, with virtual memory offering advantages such as enhanced memory security and flexibility in usage. Overall, effective memory management ensures optimal performance and resource utilization for the "Draw It or Lose It" game on the Windows operating system.

5. **Distributed Systems and Networks**: A distributed system refers to a computing environment where distinct functionalities are spread across multiple computers or computing devices interconnected via a network. For designing the "Draw It or Lose It" game, a distributed system plays a crucial role as its algorithm is distributed across two or more networks.

Employing a client-server delivery model is recommended, where each client program relies on a single server application for the game. This approach allows each application to leverage the strengths of its respective client system. A robust server network is essential, as the success of the game hinges on multiple clients connecting to a centralized server to participate in the same game instance.

To ensure uninterrupted gameplay, a distributed database system will be implemented using hubs to connect several computers via the Local Area Network (LAN) infrastructure. This setup guarantees that the game continues to run even if one machine experiences a failure. Hubs act as transmitters, amplifying weak signals that may degrade over long distances, thereby maintaining seamless connectivity and gameplay experience for all users.

6. **Security**: Security in programming entails constructing software applications in a manner that minimizes the introduction of security vulnerabilities. Ensuring the security of various operating systems is a primary concern for clients, necessitating the implementation of diverse security measures across multiple platforms.

As a foundational security step, the Windows Defender antivirus program can be employed to actively protect the client's system. Additionally, utilizing Virtual Private Network (VPN) services further enhances user data protection.

Regular and periodic security assessments are crucial to safeguard user security and prevent the unauthorized leakage of personal data. Developers must undergo adequate training in data encryption and security practices to ensure the application's robustness against security threats.

Encouraging the adoption of two-factor and multi-factor authentication mechanisms adds an extra layer of security to the client's systems. Moreover, all data transmitted between client and server should be encrypted using proven encryption methods to safeguard against unauthorized access and data breaches.

**References**

*Advantages and disadvantages of mobile technology*. (n.d.). NI Business Info. Retrieved April 19, 2024, from https://www.nibusinessinfo.co.uk/content/advantages-and-disadvantages-mobile-technology

Collins, P. (n.d.). *Game Development Stages: How Video Games Are Created?* Starloop Studios. Retrieved April 19, 2024, from https://starloopstudios.com/game-development-stages/

Sheldon, R. (2022, June 6). memory management. *TechTarget*. https://www.techtarget.com/whatis/definition/memory-management

*What languages have been used to write Windows, mac OS and linux OS?* Tutorials Point. (n.d.). Retrieved April 18, 2024, from <https://www.tutorialspoint.com/what-languages-have-been-used-to-write-windows-mac-os-and-linux-os>